Outline

• character sets (what)
• encodings (how)
• a bit intertwined
• generalizations
• Squeak specific
• european examples
Character Sets
sets of characters
ASCII

- North America, United States
- 128 characters
ISO-8859-1

(latin I)

- Western Europe
- Does not include € (ISO-8859-15 / latin-9)
- ASCII + 128 additional charters
Unicode
one character set to rule them all
Code point

“atom” of text
Part of Unicode
NOT Part of Unicode
Integer (abstract)

SmallInteger
(-1073741824
1073741823)

LargeInteger
(∞ - SmallInteger)

ranges, no endianness!
character set, not encoding!
#leadingChar

- mixes abstraction layers
- language
- presentation
Algorithms

- know Unicode
- know all the rules
- know all the code points
- know all the locales
Transformations

Fußball

FUSSBALL

locale dependent!
Collation (ordering)

- ABC...RSTUWXYZ
- ÄB...NOÖ...ßTUÜV...YZ
- ABC...RSTUWXYZÅÄÖ

locale dependent!
Normalization

(what does #= really mean?)

• ñ + a = ä

• there are different ones to chose from
PHP 6 will do all of this
Encodings

mappings from one space to another
(isomorphisms)
1:1

• ASCII
• ISO-8859-1
ASCII

- 7 bit
- 8 bit
UTF-32

16rFC 16r00 16r00 16r00 (LE)
16r00 16r00 16r00 16r00 16rFC (BE)
UTF-16

16rFC 16r00 (LE)
16r00 16rFC (BE)
WAKom

1:1 direct mapping from bytes to Charaters
WAKomEncoded*

use it with utf-8!
Content-Type: text/html; charset=utf-8

<meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
2.8: WASession >> #charSet
2.9: /seaside/config
Links

• UTF-8 Sampler
• Favourite Unicode Codepoints
• On the Goodness of Unicode
• Characters vs. Bytes
übercool